CLAIMS:

1. A method for use in a computer system having a main memory and a first processor means for providing said processor with access to the register state of a supplemental processor, said supplemental processor otherwise inaccessible by said first processor, comprising:

loading a program into said supplemental processor;

executing said program in said supplementary processor to generate said register states of said supplementary processor;

storing said register states in said main memory; and

accessing said register state by said first processor.

2. The method of claim 1 further including a program debugger operating on said main processor and wherein said method further includes:

accessing said register state in said main memory by said debugger to debug said program.

3. The method of debugging a specified program that has been operationally interrupted, nominally operating on a supplemental processor, the register states of which cannot be directly accessed by a main processor operating a debugging program comprising:

activating a secondary program in said supplemental processor to transmit register states of said supplemental processor, at the time said specified program is operationally interrupted, to said main processor;

modifying parameters of said specified program in a memory pool accessible by said main processor through the use of said debugging program in said main processor; and

restoring operation of said specified program in said supplemental processor containing alterations generated in the debugging process.

4. The method of debugging a specified program intended to operate on a supplemental processor, said supplemental processor having limited memory and flexibility, the register states of which cannot be directly accessed by a read command from another processor comprising:

reserving a pool of memory accessible to a debugging program processor;

running a specified program, to be debugged, in said supplemental processor until instructions in said specified program cause operation of said specified program to cease;

activating a secondary program in said supplemental processor to transmit register states of said supplemental processor, at the time said specified program is operationally interrupted, to said debugging program processor;

modifying parameters of said specified program in a memory pool accessible by said debugging program processor; and

restoring operation of said specified program in said supplemental processor with program alterations modified in the debugging process.

5. Debugging apparatus comprising:

a debugging processor;

a supplemental processor;

a specified program to be debugged installed for operation in said supplemental processor;

a communication path between and interconnecting said supplemental processor and said debugging processor;

a pool of memory accessible to said debugging processor; and

a debugging program operable to,

reserve a portion of said pool of memory for debugging operations,

activate a secondary program in said supplemental processor to transmit register states of said supplemental processor, at the time said specified program is operationally interrupted, to said debugging processor for insertion into said pool of memory,

modify parameters of said specified program in said pool of memory accessible by said main processor through the use of said debugging program in said debugging processor, and

restore operation of said specified program in said supplemental processor with alterations modified in the debugging process.

6. Debugging apparatus comprising:

a debugging processor;

a supplemental processor;

a specified program to be debugged installed for operation in said supplemental processor and in an operationally interrupted state;

a communication path between and interconnecting said supplemental processor and said debugging processor;

a pool of memory accessible to said debugging processor; and

a secondary program in said supplemental processor operable, on command from said debugging processor, to transmit register states of said supplemental processor, at the time said specified program is operationally interrupted, to said debugging processor for insertion into said pool of memory.

7. Debugging apparatus comprising:

a debugging processor;

a supplemental processor in communication with said debugging processor; and

a debugging program installed in said debugging processor and operable to,

reserve a portion of a pool of memory accessible to said debugging processor for debugging operations,

activate a secondary program in said supplemental processor to transmit register states of said supplemental processor to said debugging processor for insertion into said pool of memory subsequent to the time a specified program, installed therein to be debugged, is operationally interrupted,

modify parameters of said specified program in said pool of memory, and

restore operation of said specified program in said supplemental processor with alterations as created in the debugging process.

8. The method of debugging a first processor unit employing a second processor operating a debugging program comprising:

activating a secondary program in first processor to transmit register states of said first processor,

subsequent to a time a specified program to be debugged is operationally interrupted, to said second processor;

modifying parameters of said specified program in a memory pool accessible by said second processor through the use of said debugging program in said second processor; and

restoring operation of said specified program in said first processor with alterations as created in the debugging process.

9. A computer program product for debugging a first processor employing a supplemental processor, the computer program product having a medium with a debugging computer program embodied thereon, the debugging computer program comprising:

computer code for transmitting and activating a secondary program in a first processor transmitting register states of said first processor, subsequent to the operational halting of said program requiring debugging, to said second processor;

computer code for modifying parameters of said specified program in a memory pool accessible by said second processor through the use of said debugging program in said second processor; and

computer code for restoring operation of said specified program in said first processor with alterations generated by the debugging process.

10. A computer program product for authenticating code in a computer system, the computer program product having a medium with a computer program embodied thereon, the computer program comprising:

computer code for transmitting and activating a secondary program in a first processor transmitting register states of said first processor, subsequent to the operational halting of said program requiring debugging, to said second processor;

computer code for modifying parameters of said specified program in a memory pool accessible by said second processor through the use of said debugging program in said second processor; and

computer code for restoring operation of said specified program in said first processor with alterations generated by the debugging process.